

11/06/98  
Date

Express Mail Label No.:  
EM044999149US

# METHOD OF AND APPARATUS FOR PROTECTING DATA ON STORAGE MEDIUM

## AND STORAGE MEDIUM

### BACKGROUND OF THE INVENTION

#### Field of the Invention

5           The present invention relates to a method of and an  
apparatus for protecting data on a storage medium, which are  
intended to protect the data to be recorded on the storage medium  
by encrypting the data with a password, and to the storage medium  
thereof in an information processing appliance.

#### 10   Description of the Related Art

          A storage device utilizing an optical disk, a magnetic  
disk and an IC card etc, is utilized for a variety of information  
processing appliances such as a computer, a word processor and  
an electronic book etc. Information such as information  
15   related to privacy and confidential information in terms of  
duties, which should not be known by persons other than an owner,  
might be written to this type of storage device. It is required  
that the data be encrypted in order to make such information  
unknown by others.

20           FIG. 15 is an explanatory diagram showing the prior art.

          A password is set on a storage medium 90 such as an optical  
disk etc or on a storage device. When writing the data, an  
encrypting unit 91 encrypts the data with the password, and the  
encrypted data is written to the storage medium 90. Further,  
25   when reading the data, a decoding unit 92 decodes the data on  
the storage medium 90 with a password.

          Thus, a data confidentiality can be kept by encrypting

the data. In this connection, there has hitherto been a method of setting one password on the whole storage medium. There also has been a method of setting passwords different based on a file unit of the storage medium.

5 *Ins A' >* There arise, however, the problems inherent in the prior art.

10 First, as cipher texts defined as samples or combinations of the cipher texts with unencrypted plain texts become larger in quantity, the decryption by a decipherer becomes easier. A result into which the same plain text is encrypted with the same password, is equal. Therefore, when encrypted directly with the same password, a statistic characteristic of the cipher text reflects in a statistic character of the plain text.

15 Accordingly, a conventional method of encrypting with the same password on the storage medium presents such a problem that if a volume of the cipher texts is large enough to make a statistic process executable, the characteristics of the plain texts can be presumed easily.

20 Second, a large capacity storage medium such an optical disk etc is stored with the data, of which some portion such as a directory portion is structured in a fixed format. A problem peculiar to the conventional method of encrypting with the same password on the storage medium is that the password is presumed by analyzing this portion, in which case other vital  
25 data are to be deciphered.

Third, according to the conventional method of setting the password per file, when the password of some portion is

09487700-110666

5  
a' don't  
decrypted, other portions can be prevented from being decrypted. In this case, however, it is required that the different password be managed per file. This operation is troublesome and might cause a problem in which a fault such as forgetting the password and so on is easy to occur.

10 Fourth, in the large capacity exchangeable storage medium such as an optical disk etc, it is possible to carry the storage medium out and copy the storage medium. Therefore, the once-encrypted data is carried out and may be analyzed later on taking a sufficient period of time. Accordingly, the problem is that the password is easy to be presumed from the cipher text.

15 A fifth, problem is that the data has hitherto been encrypted directly with the password, and hence, if the password is changed, the whole data are required to be re-encrypted.

15  
ins a2  
SUMMARY OF THE INVENTION

20 It is a primary object of the present invention to provide a method of and an apparatus for protecting data on a storage medium, wherein a password is hard to be analyzed from a cipher text.

It is another object of the present invention to provide a method of and an apparatus for protecting data on a storage medium that are capable of changing key data with one password on a memory unit.

25 It is still another object of the present invention to provide a method of and an apparatus for protecting data on a storage medium, wherein re-encryption of data is not needed even

when a password is changed.

A method of protecting data on a storage medium according to the present invention has a write mode including a step of encrypting, after generating key data, the key data with a password and writing the encrypted key data to the storage medium, and a step of encrypting data with the key data and writing the encrypted data to the storage medium. The data protecting method also has a read mode including a step of reading the encrypted key data from the storage medium, a step of decoding the encrypted key data with the password, and a step of decoding the data on the storage medium with the decoded key data.

According to the present invention, the data is encrypted not by using the password directly as an encryption key but by using key data generated separately. The key data encrypted with the password serving as a key, and written to the storage medium. When in the reading process, the encrypted key data is decoded with the password, thereby obtaining the key data. Then, the data is decoded with the key data.

Thus, the data is encrypted by use of the key data generated separately from the password, whereby the encrypted key data is, even if a cipher text is to be analyzed, merely decrypted. The password and the key data are therefore hard to be analyzed. This makes it feasible to prevent the password from being deciphered by analyzing the cipher text.

Further, since the encryption is done using the key data generated separately from the password, a key different based

on the memory unit such as a sector etc can be imparted to one password by changing the key data. Consequently, the encryption can be accomplished by use of the key different per logic sector, whereby a data confidentiality can be enhanced.

5 Furthermore, the encryption is carried out by using the key data generated separately from the password, and therefore, even if the password is changed, the data is not required to be re-encrypted. Hence, the password can be easily changed on even a storage medium having a capacity as large as several  
10 hundred mega bytes.

Other features and advantages of the present invention will become readily apparent from the following description taken in conjunction with the accompanying drawings.

15  
BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention, and together with the  
20 general description given above and the detailed description of the preferred embodiments given below, serve to explain the principle of the invention, in which:

FIG. 1 is a block diagram showing one embodiment of the present invention;

25 FIG. 2 is a processing flowchart when in a logical formatting process in a first embodiment of the present invention;

FIG. 3 is a flowchart showing a writing process in the first embodiment of the present invention;

FIG. 4 is an explanatory diagram showing a storage region in the first embodiment of the present invention;

5 FIG. 5 is an explanatory diagram showing key data in the first embodiment of the present invention;

FIG. 6 is a flowchart showing a reading process in the first embodiment of the present invention;

10 FIG. 7 is a flowchart showing a writing process in a second embodiment of the present invention;

FIG. 8 is a flowchart showing the writing process in a third embodiment of the present invention;

FIG. 9 is an explanatory diagram showing the key data in the third embodiment of the present invention;

15 FIG. 10 is a flowchart showing the reading process in the third embodiment of the present invention;

FIG. 11 is an explanatory diagram showing a fourth embodiment of the present invention;

20 FIG. 12 is a flowchart showing a writing process in the fourth embodiment of the present invention;

FIG. 13 is a flowchart (part 1) showing a password changing process in the fourth embodiment of the present invention;

25 FIG. 14 is a flowchart (part 2) showing the password changing process in the fourth embodiment of the present invention; and

FIG. 15 is an explanatory diagram showing the prior art.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is a block diagram showing one embodiment of the present invention. FIG. 2 is a processing flowchart when in a logical format in a first embodiment of the present invention.

5 FIG. 3 is a flowchart showing a writing process in the first embodiment of the present invention. FIG. 4 is an explanatory diagram showing a storage area in the first embodiment of the present invention. FIG. 5 is an explanatory diagram showing key data in the first embodiment of the present invention. FIG.  
10 6 is a flowchart showing a reading process in the first embodiment of the present invention.

As shown in FIG. 1, a storage medium 1 is constructed of a magneto-optic disk. A logic sector size of the storage medium 1 is set to 2 KB (kilobytes). A control circuit 2 is constructed  
15 of a processor. A first encrypting unit 20 encrypts key data PS with a password PW inputted by a user, and writes encrypted key data PS' onto the storage medium 1.

A second encrypting unit 21 encrypts the data to be written with the key data PS, and writes the encrypted data onto  
20 the storage medium 1. A first decoding unit 22 decodes with the password PW the key data PS' encrypted on the storage medium 1. A second decoding unit 23 decodes the data on the storage medium 1 with the key data PS decoded, and outputs the data. A memory 3 provides an operation region of a control circuit  
25 (hereinafter referred to as a CPU). Note that the first and second encrypting units 20, 21 and the first and second decoding units 22, 23 are shown in the form of blocks as processes by

the CPU 2.

Referring to FIG. 2, a process when creating a logical format of the medium will be explained. The following processes are executed when creating the medium logical format defined  
5 as an initial process of the medium.

(S1) The user inputs the user password PW to the CPU 2.

(S2) The CPU 2 generates random numbers (each consisting of 8 bytes) for the number of sectors. This random number is defined as the key data PS. Hereinafter, the explanation is  
10 given on the assumption that "n" be the number of sectors, and random numbers PS[1] - PS[n] be generated.

(S3) The CPU 2 makes the random numbers (random data) PS[ ] (PS[1] - PS[n]) for the number of sectors stored in the operation region of the memory 3.

15 (S4) The CPU 2 encrypts each piece of key data PS[1] - PS[n] in the operation region with the password PW. As a matter of course, the whole key data PS[1] - PS[n] in the operation region may also be encrypted with the password.

(S5) The CPU 2 writes the encrypted key data PS'[1] -  
20 PS'[n] to a region L1 on the storage medium 1.

1894  
As illustrated in FIG. 4, the logical format of the storage medium (disk) 1 is shown by each sector. This sector addressed based on a logical block address LBA. Herein, in FIG. 4, there are provided X-pieces of sectors of logical block  
25 addresses LBA being [1] through [X].

The region L1 for a-sector starting from a head sector (LBA = 1) within the storage area of the optical disk, is



04  
00n  
allocated as a storage region for the encrypted key data PS' [1] - PS' [n]. namely, the number of sectors in a using region of the data is  $n (= (X-a))$ , and, per section in the using region, the encrypted key data PS' [1] - PS' [n] are stored in the region L1.

Next, a writing process to the medium will be explained with reference to FIG. 3.

(S10) It is assumed that there occurs a request for writing to a position in which the logical block address (sector number) LBA is [S0]. The sector number LBA requested is changed to [S1] so that the write request position is not overlapped with the region L1. Herein, as shown in FIG. 4, a size [a] of the region L1 is added to the sector number [S0], thereby obtaining the changed sector number [S1].

(S11) The CPU 2 judges whether or not the data (encrypted key data) in the region L1 on the optical disk 1 have already been read out. If already read out, the key data are developed in the operation region of the memory 3, and hence the processing proceeds to step S14.

(S12) The CPU 2, if the data in the region L1 have not been read out, executes a process of developing the key data in the operation region of the memory 3. Namely, the CPU 2 obtains the user password PW. Then, the CPU 2 reads the data PS' [1] - PS' [n] from the region L1 on the optical disk 1.

(S13) The CPU 2 decodes the data PS' [1] - PS' [n] in the region L1 with the password PW. The data PS' [1] - PS' [n] are thereby obtained. The data PS' [1] - PS' [n] are stored in the

operation region of the memory 3.

(S14) The CPU 2 obtains key data PS[S0] of the logical block address (sector number) LBA (= S0) out of the key data in the operation region of the memory 3. As shown in FIG. 5, the key data PS[S0] corresponding to the logical block address LBA is obtained from a key data table in the operation region of the memory 3. Then, the CPU 2 encrypts the data to be written with this piece of key data PS[S0]. An encrypting method may involve the use of known DES etc.

(S15) The CPU 2 writes the encrypted data to a position of the logical block address LBA (= S1) on the optical disk 1.

Next, the reading process will be described referring to FIG. 6.

(S20) It is presumed that there occurs a request for reading from a position where the logical block address (sector number) LBA is [S0]. The sector number LBA requested is changed to [S1] so that the read request position is not overlapped with the region L1. Herein, as shown in FIG. 4, the size [a] of the region L1 is added to the sector number [S0], thereby obtaining the changed sector number [S1].

(S21) The CPU 2 judges whether or not the data (encrypted key data) in the region L1 on the optical disk 1 have already been read out. If already read out, since the key data are developed in the operation region of the memory 3, the processing proceeds to step S24.

(S22) The CPU 2, if the data in the region L1 have not been read out, executes a process of developing the key data

in the operation region of the memory 3. Namely, the CPU 2 obtains the user password PW. Then, the CPU 2 reads the data PS'[1] - PS'[n] from the region L1 on the optical disk 1.

(S23) The CPU 2 decodes the data PS'[1] - PS'[n] in the  
5 region L1 with the password PW. The key data PS[1] - PS[n] are thereby obtained. The key data PS[1] - PS[n] are stored in the operation region of the memory 3.

(S24) The CPU 2 obtains key data PS[S0] of the logical  
10 block address (sector number) LBA (= S0) out of the key data in the operation region of the memory 3. As shown in FIG. 5, the key data PS[S0] corresponding to the logical block address LBA is obtained from the key data table in the operation region of the memory 3. Then, the CPU 2 reads in the logical block  
15 address S1 from the optical disk 1. Further, the CPU 2 decodes the thus read data with the key data PS[S0]. A decoding method may involve the use of known DES etc. The CPU 2 transmits the decoded data to a requesting terminal (e.g., a computer).

Thus, when creating the logical format of the medium, the  
random number is generated per logic sector, thereby generating  
20 the key data per logic sector. Then, the key data encrypted with the password is written to the storage medium 1. When in the data writing process, the data are encrypted with the key data and written to the storage medium 1.

When in the data reading process, the encrypted key data  
25 on the storage medium 1 are read and thereafter decoded with the password, thereby obtaining the key data. Then, the data read from the storage medium are decoded with this piece of key

data.

As described above, the data are encrypted with the key data generated separately from the password, with the result that the encrypted key data is, even if a cipher text is analyzed, merely decoded. Therefore, the password and the key data are analyzed with difficulty. This makes it possible to prevent the password from being decoded by analyzing the cipher text.

Further, the encryption is executed by using the key data generated separately from the password, and therefore a different key can be imparted based on the logic sector unit by changing the key data with respect to one password. Hence, the data can be encrypted by using the different key per logic sector, whereby the confidentiality of the data can be enhanced.

Note that the region L1 is provided in the part of the smaller logical block address but may be stored in a part of the maximum logical block address.

FIG. 7 is a flowchart showing a writing process in a second embodiment of the present invention. Referring to FIG. 7, the writing process to the medium will be explained. The process in the case of creating the logical format of the medium is executed in the same way as done in the embodiment in FIG. 2, the encrypted key data of each logic sector is stored on the storage medium 1.

(S30) It is presumed that there occurs a request for writing to a position where the logical block address (sector number) LBA is [S0]. The sector number LBA requested is changed to [S1] so that the write request position is not

overlapped with the region L1. Herein, as shown in FIG. 4, the size [a] of the region L1 is added to the sector number [S0], thereby obtaining the changed sector number [S1].

(S31) The CPU 2 judges whether or not the data (encrypted key data) in the region L1 on the optical disk 1 have already been read out. If already read out, since the key data are developed in the operation region of the memory 3, the processing proceeds to step S34.

(S32) The CPU 2, if the data in the region L1 have not been read out, executes the process of developing the key data in the operation region of the memory 3. Namely, the CPU 2 obtains the user password PW. Then, the CPU 2 reads the data PS'[1] - PS'[n] from the region L1 on the optical disk 1.

(S33) The CPU 2 decodes the data PS'[1] - PS'[n] in the region L1 with the password PW. The key data PS[1] - PS[n] are thereby obtained. The key data PS[1] - PS[n] are stored in the operation region of the memory 3.

(S34) The CPU 2 generates a random number R. Subsequently, the CPU 2 writes the random number R to the key data PS[S0] in the key data logical block address (sector number) LBA (= S0) in the operation region of the memory 3.

(S35) Then, the CPU 2 encrypts the data to be written with this piece of key data (random number R). The encrypting method may involve the use of known DES etc. The CPU 2 writes the encrypted data to a position of the logical block address LBA (= S1) on the optical disk 1.

(S36) The CPU 2 rewrites the data in the region L1 on

the optical disk 1 at a proper timing. That is, the CPU 2, if a value WC of a write counter for indicating the number of writing processes exceeds, e.g., 32, proceeds to step S37 in order to rewrite the data in the region L1 for insurance. It is the reason why the writing process is done at an interval of a predetermined number of times that a data recovery of some extent is to be compensated even if there occurs such a situation that a process of ejecting the medium can not be executed due to an occurrence of something abnormal. The numerical value such as 32 times may be arbitrary. This process is not the indispensable condition for the present invention. Further, the CPU 2, when requested to eject the storage medium 1, proceeds to step S37 in order to save the key data. Moreover, the CPU 2, of the power source is switched OFF, advances to step S37 to save the key data.

(S37) The CPU 2 encrypts each piece of the key data PS[1] - PS[n] in the operation region with the password PW. As a matter of course, the whole key data PS[1] - PS[n] in the operation region may also be encrypted with the password. Next, the CPU 2 writes the encrypted key data PS'[1] - PS'[n] to the region L1 on the storage medium 1.

In accordance with the second embodiment, in addition to the operation in the first embodiment, the different key data is generated each time the data is written. Therefore, the encryption is executed using the different key data each time the data is written, thereby enhancing the confidentiality of the data.

Note that the reading process is the same as in the first embodiment in FIG. 6, so that its explanation is omitted.

FIG. 8 is a flowchart showing a writing process in a third embodiment of the present invention. FIG. 9 is an explanatory diagram showing the key data in the third embodiment of the present invention. FIG. 10 is a flowchart showing the reading process in the third embodiment of the present invention.

1ns05  
09187700-110698  
When in a medium logical formatting process, as in the first embodiment shown in FIG. 2, the region L1 on the optical disk 1 is stored with encrypted key data PS'[1] - PS'[512]. Herein, however, the encrypted data is not stored per logic sector. For example, it is assumed that a capacity of the region L1 be 4 KB. Then, supposing that the password be an 8-byte/entry, as shown in FIG. 9, 512-pieces of key words (entries) PS[1] - PS[512] are generated. Subsequently, the region L1 is stored with the 512-pieces of encrypted key words PS'[1] - PS'[512].

The writing process is described with reference to FIG. 8.

(S40) It is presumed that there occurs a request for writing to a position where the logical block address (sector number) LBA is [S0]. The sector number LBA requested is changed to [S1] so that the write request position is not overlapped with the region L1. Herein, as shown in FIG. 4, the size [a] of the region L1 is added to the sector number [S0], thereby obtaining the changed sector number [S1].

(S41) The CPU 2 judges whether or not the data (encrypted key data) in the region L1 on the optical disk 1 have already

been read out. If already read out, since the key data are developed in the operation region of the memory 3, the processing proceeds to step S44.

(S42) The CPU 2, if the data in the region L1 have not  
5 been read out, executes the process of developing the key data in the operation region of the memory 3. Namely, the CPU 2 obtains the user password PW. Then, the CPU 2 reads the data PS'[1] - PS'[n] from the region L1 on the optical disk 1.

(S43) The CPU 2 decodes the data PS'[1] - PS'[n] in the  
10 region L1 with the password PW. The key data PS[1] - PS[n] are thereby obtained. The key data PS[ ] (PS[1] - PS[n]) are stored in the operation region of the memory 3.

(S44) The CPU 2 obtains four values R0, R1, R2, R3 from  
the sector number S0 requested. Herein, the logic sector number  
15 S0 is assumed to be a bit string of 32 bits, and 32 bits are rearranged by 8 bits to the individual values R0, R1, R2, R3. R0 - R3 take values above 0 but less than 256. Then, with R0 - R3 serving as an index, random number values (key data) are taken out of PS[ ] in the operation region of the memory 3. Based  
20 on the thus taken-out four values, an 8-byte random number (key data) R is generated.

Herein, as shown in FIG. 9, the key data PS[R0]  
corresponding to R0 is taken out, and key data [R1 + 256]  
corresponding to (R1 + 256) is taken out. Key data PS [R2 +  
25 256] corresponding to R2 is taken out, and key data PS [R3]  
corresponding to R3 is taken out.

Then, the key data R is calculated by the following



arithmetic formula:

$$R = (PS[R0] * PS[R1 + 256]) * (PS[R2 + 256] + PS[R3])$$

where [\*] represents an EOR calculation.

(S45) Then, the CPU 2 encrypts the data to be written with  
5 this piece of key data R. The encrypting method may involve  
the use of known DES etc. The CPU 2 writes this piece of  
encrypted data to a position of the logical block address LBA  
(= S1) on the optical disk 1.

Next, the reading process is explained with reference to  
10 FIG. 10.

(S50) It is presumed that there occurs a request for  
reading from a position where the logical block address (sector  
number) LBA is [S0]. The sector number LBA requested is  
changed to [S1] so that the read request position is not  
15 overlapped with the region L1. Herein, as shown in FIG. 4, the  
size [a] of the region L1 is added to the sector number [S0],  
thereby obtaining the changed sector number [S1].

(S51) The CPU 2 judges whether or not the data (encrypted  
key data) in the region L1 on the optical disk 1 have already  
20 been read out. If already read out, the key data are developed  
in the operation region of the memory 3, and hence the processing  
proceeds to step S54.

(S52) The CPU 2, if the data in the region L1 have not  
been read out, executes the process of developing the key data  
25 in the operation region of the memory 3. Namely, the CPU 2  
obtains the user password PW. Then, the CPU 2 reads the data  
PS'[1] - PS'[n] from the region L1 on the optical disk 1.

(S53) The CPU 2 decodes the data  $PS'[1] - PS'[n]$  in the region L1 with the password PW. The key data  $PS[1] - PS[n]$  are thereby obtained. The key data  $PS[ ]$  ( $PS[1] - PS[n]$ ) are stored in the operation region of the memory 3.

5 (S54) The CPU 2 obtains the four values  $R0, R1, R2, R3$  from the sector number  $S0$  requested. Herein, the logic sector number  $S0$  is assumed to be a bit string of 32 bits, and 32 bits are rearranged by 8 bits to the individual values  $R0, R1, R2, R3$ .  $R0 - R3$  take values above 0 but less than 256. Then, with  
10  $R0 - R3$  serving as an index, random number values (key data) are taken out of  $PS[ ]$  in the operation region of the memory 3. Based on the thus taken-out four values, the 8-byte random number (key data)  $R$  is generated.

Herein, as shown in FIG. 9, the key data  $PS[R0]$   
15 corresponding to  $R0$  is taken out, and the key data  $[R1 + 256]$  corresponding to  $(R1 + 256)$  is taken out. The key data  $PS [R2 + 256]$  corresponding to  $R2$  is taken out, and the key data  $PS [R3]$  corresponding to  $R3$  is taken out.

Then, the key data  $R$  is calculated by the above-mentioned  
20 arithmetic formula.

(S55) Then, the CPU 2 reads the data in the logical block address  $LBA (= S1)$  from the optical disk 1. Further, the CPU 2 decodes the read data with this piece of key data  $R$ . The decoding method may involve the use of known DES etc.

Insat  
~~The size of the region L1 on the optical disk 1 can be made smaller in the third embodiment than in the first embodiment. Namely, it is required in the first embodiment, that~~

09187700-110698

96  
CDN-4

the same number of pieces of key data as the number of the logic sectors be stored. For instance, supposing that one sector be 2 KB, the storage capacity be 600 MB and the key data be 8 bytes, the region L1 is required to have a capacity of 2.4 MB. In the third embodiment, 512-pieces of key data are stored, and therefore approximately 4 KB may suffice for the region L1.

Besides, even with such settings, the random number is generated based upon the calculation, and hence the key data different per sector can be obtained.

10 FIG. 11 is an explanatory diagram showing a fourth embodiment of the present invention. FIG. 12 is a flowchart showing the writing process in the fourth embodiment of the present invention.

15 The fourth embodiment shows, in addition to what has been shown in the third embodiment, a method capable of using a plurality of user passwords. As shown in FIG. 11, an n-number of users are accepted, and therefore passwords PW1 - PWN are set per user. On the assumption that the password consists of 8 bytes, corresponding to each user, the optical disk 1 is provided with 8-byte (a size of PW1) regions L2 - Ln and 8-byte regions C1 - Cn.

25 When creating the logical format of the storage medium, as in the third embodiment, what is obtained by encrypting the random number data with the user password PW1 is written to the region L1.

In addition, an authentication character string DC1 of the password is generated, and what is obtained by encrypting

the character string DC1 with the password PW1 is written to the region C1. Further, what is obtained by encrypting the password PW1 with PW2 is written to the region L2, and what results from encryption of the password PW1 with PWN is written  
5 to the region Ln.

Moreover, what is acquired by encrypting an authentication character string DC2 of the password PW2 with the password PW2, is written to the region C2. Hereinafter, what results from encryption of the authentication character  
10 string DCn of the password PWN with the password PWN, is written to the region Cn.

The authentication character string of each password serves to authenticate whether the inputted password is correct or not. This authentication character string may be structured  
15 of a confidential character string peculiar to the system or composed of a value (e.g., an exclusive OR of a password PWi and a certain specified character string) calculated from the password PWi.

Next, the data writing/reading process in the case of  
20 using the user password is executed in the same way with the third embodiment shown in FIGS. 8 and 10.

The data writing process when using the user password PWi ( $i > 1$ ) is explained referring to FIG. 12.

(S60) It is presumed that there occurs a request for  
25 writing to a position where the logical block address (sector number) LBA is [S0]. The sector number LBA requested is changed to [S1] so that the write request position is not

overlapped with the region L1. Herein, as shown in FIG. 4, the size [a] of the region L1 is added to the sector number [S0], thereby obtaining the changed sector number [S1].

(S61) The CPU 2 judges whether or not the data (encrypted key data) in the region L1 on the optical disk 1 have already been read out. If already read out, the key data are developed in the operation region of the memory 3, and hence the processing proceeds to step S64.

Ins07  
09187700 110696  
10 (S62) The CPU 2, if the data in the region L1 have not been read out, executes the process of developing the key data in the operation region of the memory 3. Namely, the CPU 2 obtains the user password PWi. Then, the CPU 2 reads the data from a region Li and decodes the read data with the password PWi. The password PW1 is thereby obtained. on the optical disk  
15 1.

(S63) Next, the CPU 2 reads the data PS' [1] - PS' [n] from the region L1 on the optical disk 1. The CPU 2 decodes the data PS' [1] - PS' [n] in the region L1 with the password PW1. The key data PS[1] - PS[n] are thereby obtained. The key data PS[ ]  
20 (PS[1] - PS[n]) are stored in the operation region of the memory 3.

(S64) The CPU 2 obtains four values R0, R1, R2, R3 from the sector number S0 requested. Herein, the logic sector number S0 is assumed to be a bit string of 32 bits, and 32 bits are  
25 rearranged by 8 bits to the individual values R0, R1, R2, R3. Then, with R0 - R3 serving as an index, the random number values (key data) are taken out of PS[ ] in the operation region of

the memory 3. Based on the thus taken-out four values, the 8-byte random number (key data) R is generated.

Herein, as shown in FIG. 9, the key data PS[R0] corresponding to R0 is taken out, and the key data [R1 + 256] corresponding to (R1 + 256) is taken out. The key data PS [R2 + 256] corresponding to R2 is taken out, and the key data PS [R3] corresponding to R3 is taken out.

Then, the key data R is calculated by the above-described arithmetic formula.

10 (S65) Then, the CPU 2 encrypts the data to be written with this piece of key data R. The encrypting method may involve the use of known DES etc. The CPU 2 writes this piece of encrypted data to a position of the logical block address LBA (= S1) on the optical disk 1.

15 Thus, the plurality of user passwords can be used.

FIG. 13 is a flowchart (part 1) showing a password changing process in a fourth embodiment of the present invention. FIG. 14 is a flowchart (part 2) showing the password changing process in the fourth embodiment of the present invention.

20 In the construction shown in FIG. 11, a process of changing the user password PW1 will be explained with reference to FIG. 13.

(S70) The CPU 2 obtains the old password PW1 and a new password PW1' as well.

25 (S71) The CPU 2 reads the data from the regions L1 and C1 on the optical disk 1.

(S72) The CPU 2 decodes the encrypted key data in the

region L1 with the password PW1, thereby obtaining the key data PS[ ]. Then, the CPU 2 decodes the data in the region C1 with the password PW1. Further, the CPU 2 judges from the decoded authentication character string whether the password PW1 is  
5 valid or not. If the password is invalid, there must be, it is conceived, an error.

(S73) The CPU 2 encrypts the key data PS[ ] with the new password PW1' and writes the encrypted data to the region L1 on the optical disk 1.

10 (S74) Next, the CPU 2 creates an authentication character string DC1' with respect to the new password PW1'. Then, the CPU 2 encrypts the authentication character string DC1' with the new password PW1', thereby obtaining a write value C1'. Further, the CPU 2 writes the write value C1' to the region C1  
15 on the optical disk 1.

The old password, of which the validity is thus confirmed, can be changed to the new password. Besides, the password can be changed without any necessity for re-encryption of the data. This method is effective in the case of the user password being  
20 single.

In the case of setting the plurality of user passwords, when changed to the new password by executing the process in FIG. 13, no data access can be done by using the user passwords PW2 - PWn. If this is inconvenient when the plurality of user  
25 passwords PW2 - PWn are set, the password PW1 is not used as the user password, and only the user password PWi (i > 1) may be used as the user password.

09187700-110698

A process of changing this user password  $PW_i$  ( $i > 1$ ) is explained with reference to FIG. 14.

(S80) The CPU 2 obtains the old password  $PW_i$  and a new password  $PW_i'$  as well.

5       (S81) The CPU 2 reads the data from the regions  $Li$  and  $Ci$  on the optical disk 1.

10       (S82) The CPU 2 decodes the encrypted key data in the region  $Li$  with the password  $PW_i$ , thereby obtaining the password  $PW_1$ . Then, the CPU 2 decodes the data in the region  $Ci$  with the password  $PW_i$ . Further, the CPU 2 judges from the decoded authentication character string whether the password  $PW_i$  is valid or not. If the password is invalid, there must be, it is conceived, an error.

15       (S83) The CPU 2 encrypts the password  $PW_1$  with the new password  $PW_i'$  and writes the encrypted data to the region  $Li$  on the optical disk 1.

20       (S84) Next, the CPU 2 creates the authentication character string  $DCi'$  with respect to the new password  $PW_i'$ . Then, the CPU 2 encrypts the authentication character string  $DCi'$  with the new password  $PW_i'$ , thereby obtaining a write value  $Ci$ . Further, the CPU 2 writes the write value  $Ci'$  to the region  $Ci$  on the optical disk 1.

25       The old password  $PW_i$ , of which the validity is thus confirmed, can be changed. In this embodiment also, the password can be changed without any necessity for re-encryption of the data.

Other than the embodiment discussed above, the present



invention can be modified as follows:

(1) The storage medium has been explained so far in the form of the magneto-optic disk, and other applicable storage mediums may be an optical disk, a magnetic disk and an IC card etc.

5 (2) The arithmetic formula for obtaining the random number R may include an application of other arithmetic formulae.

The present invention has been discussed so far by way of the embodiments but may be modified in a variety of forms within the range of the gist of the present invention, and those modifications are not excluded from the scope of the present invention.

As discussed above, the present invention exhibits the following effects.

15 (1) The data is encrypted not by using the password directly as the encryption key but by using the key data generated separately from the password. Even if the cipher text is analyzed, the encrypted key data is merely decoded. Therefore, the password and the key data are analyzed with  
20 difficulty. This makes it possible to prevent the password from being decoded by analyzing the cipher text.

(2) Further, the encryption is executed by using the key data generated separately from the password, and therefore a different key can be imparted based on the logic sector unit  
25 by changing the key data with respect to one password. Hence, the data can be encrypted by using the different key per logic sector, whereby the confidentiality of the data can be enhanced.

(3) Moreover, the data is encrypted by use of the key data generated separately from the password, and therefore, even when the password is changed, the re-encryption of the data is not required. Hence, the password can be easily changed with  
5 respect to the storage medium having a capacity as large as several hundred mega bytes.

09187700 110698